

UNITED STATES PATENT APPLICATION

of

JOE F. BRITT, JR.

ANDREW T. MCFADDEN

PHILLIP Y. GOLDMAN

and

BRUCE A. LEAK

for

DOWNLOADING SOFTWARE

FROM A SERVER TO A CLIENT

1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE

BACKGROUND OF THE INVENTION

1. Related Applications

This application is a continuation of U.S. Patent Application Serial No. 09/437,887, filed November 9, 1999, which is a continuation-in-part of U.S. Patent Application Serial No. 09/106,937, filed June 30, 1998, now issued as U.S. patent No. 6,023,268, which is a divisional of U.S. Patent Application Serial No. 08/755,238, filed November 22, 1996, now issued as U.S. Patent No. 5,940,074, which is a continuation-in-part of U.S. Patent Application Serial No. 08/660,088, filed June 3, 1996, now issued as U.S. Patent No. 6,034,689, which is a co-pending application with U.S. Patent Application Serial No. 08/656,924, filed June 3, 1996, now issued as U.S. Patent No. 5,918,013. The foregoing patents and patent applications are incorporated herein by reference.

2. The Field of the Invention

The present invention pertains to the field of computer software. More particularly, the present invention relates to a system and method for remotely upgrading software over a network.

3. The Prior State of the Art

Computer software has proven to be a valuable tool for mankind. Software includes computer-executable instructions which, when executed, perform one or more predetermined functions. There are two major categories of software applications, system software and application software. System software includes the basic software needed to control a computer. Such system software may include, for example, an operating system and a database manager. Applications software uses services provided by the system

1 software, and also provides its own methods to process data for a user. For example,
2 applications software might perform word processing, spreadsheet calculation, flight
3 simulation, or any other conceivable function or group of functions.

4 A problem encountered by computer users in general is that software applications
5 tend to become outdated quickly. Accordingly, software suppliers periodically produce
6 upgrades, which are often distributed in the same way that the original software was
7 distributed, such as on magnetic or optical disks or other similar storage devices. However,
8 the distribution of software upgrades on storage media such as these has disadvantages. For
9 example, it is inconvenient and sometimes annoying for the user to have to repeatedly install
10 software upgrades, which can be a time-consuming process. Further, a user may not be
11 aware that an upgrade is available or necessary, or the user may forget to obtain or install the
12 upgrade. The failure or delay in installing an upgrade can be detrimental since the upgrade
13 may add valuable new features to the software or remedy a "bug" (i.e., error) in the
14 software. Therefore, what is desired is a technique for allowing a software upgrade to be
15 automatically provided over a network in a manner which requires little or no effort on the
16 part of the user.

17 Another problem encountered by computer users is that sometimes software or data
18 files becomes corrupted. A "corrupted" file is a file that has been changed presumably
19 unintentionally by a failure in either hardware or software, causing the bits within the file to
20 be altered so as to render the file unreadable as intended. Typically, the corrupted file can
21 only be fixed by re-installing the software, or recovering the data from a backup. Re-
22 installing the software can be time consumer and annoying to the user. Furthermore,
23 recovering of data also conventionally requires extensive user interaction. Also, the user
24 may not have the knowledge or capability to reinstall the software or recover the data.

1 Therefore, what is also desired is a technique for fixing a corrupted file which requires little
2 or no effort on the part of the user.

3 4 SUMMARY OF THE INVENTION

5 The present invention relates to upgrading or restoring software stored on a client,
6 such as a computer or set-top box. In one implementation of the invention, inoperable or
7 corrupted software at the client is automatically restored. During power up or at any other
8 predetermined time, the client automatically checks whether the software or data is corrupt
9 or valid (i.e., not corrupt) using, for example, a conventional checksum technique. If the
10 client determines that the software or data is corrupt, the client automatically connects to a
11 server that contains a replacement for the corrupt software or data. Then, the replacement is
12 downloaded over the network infrastructure to the client so that the corrupt software or data
13 can be replaced by the valid software or data, respectively. Thus, corrupt software or data
14 can be automatically fixed with little, if any, effort required of the user.

15 In another implementation of the invention, a software upgrade is transmitted to the
16 client via a satellite link. The software upgrade occurs automatically with little if any input
17 required from a user. Thus, the upgrade is much more convenient to the user than the
18 traditional methods of upgrading software using a magnetic or optical disk that contains the
19 upgrade. Furthermore, since the upgrade is automatic, there is no risk that the upgrade will
20 be forgone because the user has forgotten or is unaware of the upgrade. Thus, the user can
21 easily and consistently have access to the most recent version of the software available.

22 As soon as one of the plurality of servers accesses a software upgrade corresponding
23 to the software stored on the client, the server transmits the upgrade to the client over a
24 satellite link. Since the download can occur via satellite transmission, the client need not be

1 linked to or actively communicating over the network infrastructure for the download to
2 occur. Thus, the server might download the upgrade during a low traffic time such as during
3 the night. When the client next begins operation, the client already has the upgrade of the
4 software without the user having done anything.

5 The present invention may be implemented in a networked computer system having
6 servers connected to a network infrastructure. This network infrastructure may be a local-
7 area network such as an Ethernet or Token ring, or may be a wide-area network such as the
8 Internet. The software to be upgraded or restored is located on a client also connected to the
9 network infrastructure.

10 The client system may be, for example, a WebTV™ system which includes a
11 television monitor on which to display information retrieved over the network infrastructure.
12 In the case of the Internet, this information might include, for example, a Web page. The
13 server that transmitted the upgrade or replacement software may be, for example, a
14 WebTV™ server. In this case, the upgrade or replacement software may be an Internet
15 browser.

16 Additional objects and advantages of the invention will be set forth in the description
17 which follows, and in part will be obvious from the description, or may be learned by the
18 practice of the invention. The objects and advantages of the invention may be realized and
19 obtained by means of the instruments and combinations particularly pointed out in the
20 appended claims. These and other objects and features of the present invention will become
21 more fully apparent from the following description and appended claims, or may be learned
22 by the practice of the invention as set forth hereinafter.

1 **BRIEF DESCRIPTION OF THE DRAWINGS**

2 In order that the manner in which the above-recited and other advantages and objects
3 of the invention are obtained, a more particular description of the invention briefly described
4 above will be rendered by reference to specific embodiments thereof which are illustrated in
5 the appended drawings. Understanding that these drawings depict only typical embodiments
6 of the invention and are not therefore to be considered limiting of its scope, the invention
7 will be described and explained with additional specificity and detail through the use of the
8 accompanying drawings in which:

9 Figure 1 illustrates several WebTV[®] client systems connected to a WebTV[®] server
10 system;

11 Figure 2 illustrates a WebTV[®] server system;

12 Figure 3 illustrates a WebTV[®] client system;

13 Figure 4 is a block diagram of an electronics unit of a WebTV[®] client system;

14 Figure 5 illustrates the functional relationship between hardware and software in the
15 client processing system of Figure 1;

16 Figure 6 is a flow diagram illustrating a reset routine for a WebTV[®] client system;

17 Figure 7 is a flow diagram illustrating a normal start-up routine for initiating an
18 upgrade of a WebTV[®] client system over the network;

19 Figure 8 is a flow diagram illustrating a routine for initiating a download for the
20 purpose of upgrading a WebTV[®] client system;

21 Figure 9 is a flow diagram depicting a routine for transmitting software upgrades to
22 clients via a satellite link.

23 Figure 10 is a flow diagram illustrating a routine for initiating a download to correct
24 an error; and

1 Figure 11 is a flow diagram illustrating a routine for downloading to the WebTV®
2 system from the network.
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

1 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

2 The present invention relates to techniques for obtaining and maintaining operable
3 and updated versions of software at a client connected to a network. In a first embodiment,
4 the client, which may be a computer or set-top box, detects the presence of corrupted
5 software and automatically connects to a server that can provide a non-corrupted copy of the
6 software. In a second embodiment, a satellite link in a network is used to download a new
7 version, or an upgrade, of software stored at the computer or set-top box..

8 In the following description, for purposes of explanation, numerous specific details
9 are set forth in order to provide a thorough understanding of the present invention. It will be
10 evident, however, to one skilled in the art that the present invention may be practiced
11 without these specific details. In other instances, well-known structures and devices are
12 shown in block diagram form in order to facilitate description.

13 Embodiments within the scope of the present invention include computer-readable
14 media having computer-executable instructions or data structures stored thereon. Such
15 computer-readable media can be any available media which can be accessed by a general
16 purpose or special purpose computer such as a set-top box. By way of example, and not
17 limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM
18 or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any
19 other medium which can be used to store the desired computer-executable instructions or
20 data structures and which can be accessed by a general purpose or special purpose computer.

21 When information is transferred or provided over a network or another
22 communications connection to a computer, the computer properly views the connection as a
23 computer-readable medium. Thus, such a connection is also properly termed a computer-
24 readable medium. Combinations of the above should also be included within the scope of

1 computer-readable media. Computer-executable instructions comprise, for example,
2 instructions and data which cause a general purpose computer, special purpose computer, or
3 special purpose processing device such as a set-top box to perform a certain function or
4 group of functions.

5 Although not required, the invention will be described in the general context of
6 computer-executable instructions, such as program modules, being executed by computers
7 or set-top boxes in network environments. Generally, program modules include routines,
8 programs, objects, components, data structures, etc. that perform particular tasks or
9 implement particular abstract data types. Computer-executable instructions, associated data
10 structures, and program modules represent examples of the program code means for
11 executing steps of the methods disclosed herein.

12 Those skilled in the art will appreciate that the invention may be practiced in
13 network computing environments with many types of computer system configurations,
14 including set-top boxes, personal computers, hand-held devices, multi-processor systems,
15 microprocessor-based or programmable consumer electronics, network PCs, minicomputers,
16 mainframe computers, and the like. The invention may also be practiced in distributed
17 computing environments where tasks are performed by local and remote processing devices
18 that are linked through a communications network. In a distributed computing environment,
19 program modules may be located in both local and remote memory storage devices.

20 In one embodiment, the present invention is implemented in a set-top box that is
21 connected to a television and to one or more servers over the Internet. The set-top box
22 includes a processing system that executes browser software to enable a user to browse
23 through World-Wide Web pages displayed on the television using a remote control device.
24 As will be described below, the browser is upgraded or reconfigured by downloading to the

1 set-top box replacement software or data transmitted from a server over the satellite network
2 and then writing the replacement software or data into a programmable, non-volatile
3 memory in the set-top box.

4 For example, the present invention may be included in a system known as WebTV[®],
5 which uses a standard television set as a display device for browsing the Web and which
6 connects to a conventional network, such as the Internet, using standard telephone, ISDN, or
7 similar communication lines. In accordance with the present invention, a user of a WebTV[®]
8 client system can utilize WebTV[®] network services provided by one or more remote
9 WebTV[®] servers. The WebTV[®] network services are used in conjunction with software
10 running in a WebTV[®] client system to browse the Web, send electronic mail, and to make
11 use of the Internet in various other ways. The WebTV[®] servers function as proxies by
12 retrieving, from a remote server, Web pages or other data requested by a WebTV[®] client
13 system and then transmitting the requested information to the WebTV[®] client system.
14 Although the invention is described herein largely in the context of WebTV[®] components,
15 the invention can be practiced with any other networks of components that are adapted to
16 perform the functions disclosed herein. Thus, any reference to WebTV components,
17 networks, services, etc. also apply to any other analogous components, networks, and
18 services.

19 20 I. System Overview

21 Figure 1 illustrates a configuration of a WebTV[®] network with which the invention
22 can be practiced according to one embodiment. A number of WebTV[®] client systems 1
23 (hereinafter, "client 1") are coupled to a modem pool 2 via direct-dial, bi-directional data
24 connections 29, which may be telephone (POTS, i.e., "plain old telephone service"), ISDN

1 (Integrated Services Digital Network), T-1, or any other similar type of connection. The
2 modem pool 2 is a conventional modem pool, such as those provided by Internet Service
3 Providers (ISPs). The modem pool 2 is coupled typically through a router, such as that
4 conventionally known in the art, to a number of remote servers 4 via a conventional network
5 infrastructure 3, such as the Internet (hereinafter, "the Internet 3").

6 The WebTV® system also includes a WebTV® server system 5 (hereinafter, "the
7 server 5") which specifically supports the clients 1 by, for example, providing the WebTV®
8 services to the clients 1. Each of the clients 1 can communicate with the server 5 either
9 directly or indirectly through the Internet 3 via the modem pool 2 using connections 29. In
10 addition, information may be downloaded from the server 5 to the client 1 using a
11 conventional satellite link 29'.

12 13 A. Server System Architecture

14 The server 5 includes one or more computer systems generally having the
15 architecture illustrated in Figure 2. It should be noted that the illustrated architecture is only
16 exemplary; the server 5 is not constrained to the illustrated architecture. The illustrated
17 architecture includes a central processing unit (CPU) 50, read-only memory (ROM) 51,
18 random access memory (RAM) 52, a mass storage device 53, a modem 54, a network
19 interface card (NIC) 55, a satellite transmitter 56, and various other input/output (I/O)
20 devices 57. Mass storage device 53 includes a magnetic, optical, or other equivalent storage
21 medium. I/O devices 57 may include any or all of devices such as a display monitor,
22 keyboard, cursor control device, etc. Modem 54 is used to communicate data to and from
23 remote servers 4 via the Internet. The satellite transmitter 57 is used to transmit information
24 to the client 1 over the satellite link 29'.

As noted above, the server 5 may actually comprise multiple physical and logical devices connected in a distributed architecture. Accordingly, NIC 55 is used to provide data communication with other devices that are part of the WebTV® services. Modem 54 may also be used to communicate with other devices that are part of the WebTV® services and which are not located in close geographic proximity to the illustrated device.

B. Client System Architecture

Figure 3 illustrates a client 1 with which the invention can be practiced according to one embodiment. The client 1 includes an electronics unit 10 (hereinafter referred to as "the WebTV® box 10" or "the box 10"), an ordinary television monitor 12, and a hand-held remote control 11. In an alternative embodiment of the present invention, the box 10 is built into the television set 12 as an integral unit. The box 10 includes hardware and software for providing the user with a graphical user interface, by which the user can access the WebTV® network services, browse the Web, send e-mail, and otherwise access the Internet.

The client 1 uses the television set 12 as a display device and an audio output device. The box 10 is coupled to the television set 12 by a link 6. The link 6 includes an audio channel for generating sound from the television's speaker and a video channel in the form of RF (radio frequency), S-video, composite video, or other format. The communication link 29 between the box 10 and the server 5 is either a telephone (POTS) connection 29a, an ISDN connection 29b, or any other type of terrestrial link. Connections 29a and 29b represent examples of two-way communications links. In addition, Figure 3 illustrates a satellite link 29', which is a one-way communication link to the client 1. The box 10 receives AC (alternating current) power through an AC power line 7.

1 Remote control 11 is operated by the user in order to control the client 1 to browse
2 the Web, send e-mail, and perform other Internet-related functions. The box 10 receives
3 commands from remote control 11 via an infrared (IR) communication link. In alternative
4 embodiments, the link between the remote control 11 and the box 10 may be RF or any
5 equivalent mode of transmission.

6 The box 10 includes application software which, when executed by a processor in
7 the box 10, provides the user with a graphical user interface by which the user can access the
8 WebTV® network services and browse the Web. The application software is automatically
9 executed upon application of power to the box 10.

10 Figure 4 shows the internal components of the box 10. Operation of the client 1 is
11 controlled by a CPU 21, which is coupled to an Application-Specific Integrated Circuit
12 (ASIC) 20. The CPU 21 executes software designed to implement features of the present
13 invention. ASIC 20 contains circuitry which is used to implement certain functions of the
14 WebTV® system. ASIC 20 is coupled to an audio digital-to-analog converter 25 which
15 provides audio output to television 12. In addition, ASIC 20 is coupled to a video encoder
16 26 which provides video output to television set 12. An IR interface 24 detects IR signals
17 transmitted by remote control 11 and, in response, provides corresponding electrical signals
18 to ASIC 20. A standard telephone modem 27 and an ISDN modem 30 are coupled to ASIC
19 20 to provide connections 29a and 29b, respectively, to the modem pool 2 and to the server
20 5. Note that, while the illustrated embodiment includes both a telephone modem 27 and an
21 ISDN modem 30, either one of these devices will suffice to practice the present invention.
22 Further, in various other embodiments, the telephone modem 27 and the ISDN modem 30
23 each may be replaced by or supplemented with other communications devices, such as a
24 cable television modem. In addition, in other embodiments, communication with the server

1 5 might be made via a token ring or Ethernet connection. Note that the box 10 also may
2 include a cable television modem (not shown). A satellite receiver 33 is coupled to the
3 ASIC 20 to receive data over the satellite link 29'.

4 Also coupled to ASIC 20 is mask Read-Only Memory (ROM) 22a, a flash memory
5 22b, and a Random Access Memory (RAM) 23. Mask ROM 22a (which is so named
6 because it is non-programmable) provides storage of certain program instructions and data,
7 as will be described below. Flash memory 22b is a conventional flash memory device that
8 can be written to (programmed) and erased electronically. Flash memory 22b provides
9 storage of the browser software as well as data. In one embodiment, a mass storage device
10 28 is included in the box 10 and coupled to ASIC 20. The mass storage device 28 may be
11 used to input software or data to the client or to download software or data received over
12 connection 29 or 29'. The mass storage device 28 includes any suitable medium for storing
13 machine-executable instructions, such as magnetic disks, optical disks, and the like.

14 As mentioned above, the box 10 includes application software including a Web
15 browser. Referring now to Figure 5, the above-mentioned application software 31 operates
16 in conjunction with operating system (OS) software 32. The OS software 32 includes
17 various device drivers and otherwise provides an interface between the application software
18 31 and the system hardware components 40 (i.e., the elements illustrated in Figure 4).

19 In one embodiment, the application software 31 and the OS software 32 are stored in
20 flash memory 22b. It will be recognized, however, that some or all of either the application
21 software 31 or the OS software 32 or both can be stored in any other suitable storage
22 medium, such as mask ROM 22a or mass storage device 28, in various embodiments.

23 As mentioned above, steps according to the present invention are embodied in
24 machine-executable instructions according to one embodiment. For example, in one

1 embodiment, the present invention is carried out in the WebTV® box 10 by the CPU 21
2 executing sequences of instructions contained in mask ROM 22a, flash memory 22b, or
3 RAM 23, or a combination of these devices. More specifically, execution of the sequences
4 of instructions causes the CPU 21 to perform the steps of the present invention. These steps
5 will be described below.

6 Instructions for carrying out the present invention may be loaded into memory from
7 a persistent store, such as mass storage device 28, and/or from one or more other computer
8 systems over a network, such as the server 5 or a remote server 4. For example, such a
9 server system may transmit a sequence of instructions to the client 1 in response to a
10 message transmitted to the server system over the Internet 3 by the client 1. As the client 1
11 receives the instructions via a network connection, such as modem 27, modem 30, or
12 satellite receiver 33, the client 1 stores the instructions in a memory. The client 1 may store
13 the instructions for later execution or execute the instructions as they arrive over the network
14 connection.

15 In some embodiments, the downloaded instructions may be directly supported by the
16 CPU 21. Consequently, execution of the instructions may be performed directly by the CPU
17 21. In other embodiments, the instructions may not be directly executable by the CPU 21.
18 Under these circumstances, the instructions may be executed by causing the CPU 21 to
19 execute an interpreter that interprets the instructions, or by causing the CPU 21 to execute
20 instructions which convert the received instructions into instructions that can be directly
21 executed by the CPU 21.

22 Certain embodiments and aspects of the present invention may be carried out in the
23 server 5, instead of (or in addition to) being carried out in the client 1. For example, the
24

1 CPU 50 of the WebTV server 5 may execute instructions stored in memory to perform steps
2 in accordance with the present invention.

3 In various embodiments, hardwired circuitry may be used in place of, or in
4 combination with, software instructions to implement the present invention. Thus, the
5 present invention is not limited to any specific combination of hardware circuitry and
6 software, nor to any particular source for the instructions executed by a computer system.

8 II. Remote Download

9 As indicated above, the browser software can be upgraded, reconfigured, or restored
10 by downloading to the box replacement software or data transmitted from a server via the
11 Internet, a direct phone connection, or via a satellite link 29'. Figures 6-11 illustrate various
12 techniques of upgrading and restoring software at the client.

13 The programmable nature of flash memory 22b and its ability to retain programmed
14 information in the absence of power are used to advantage in performing such an upgrade or
15 reconfiguration. As will be explained in greater detail below, replacement software or data
16 transmitted from the server 5 (or another server designated by the server 5) is automatically
17 written into the flash memory 22b in the box 10. In addition, the flash memory can be used
18 to store various resources downloaded from the Web, such as Java applets (programs), so
19 that such resources will be retained in the event of loss of power to the client 1. Note that
20 the present invention does not necessarily require use of a flash memory for these purposes;
21 other forms of programmable non-volatile memory may be used, such as an electrically-
22 erasable programmable ROM (EEPROM).

23 A download from the Internet 3 can be performed for purposes of upgrading the
24 client 1 or reconfiguring the client 1 to correct an inconsistent state or corruption (e.g., an

1 error in the program instructions or data). The client 1 determines whether a download from
2 the network should take place each time the client 1 is reset. More specifically, a download
3 will be performed if, upon reset, the client system 1 detects an error condition or, in the case
4 of an upgrade, the client system 1 finds a download request in memory. If the satellite link
5 29' is available, the server 5 might initiate a download to download browser software
6 updates as they become available, without requiring a request from the client 1.

7 Figure 6 illustrates a method 600 performed by the client for restoring corrupted
8 software according to a first embodiment of the invention. In this embodiment, the method
9 may be embodied in a reset routine that is performed any time the client 1 is reset during
10 operation or any time the box 10 is powered on. For example, the client system 1 may
11 perform the method 600 of Figure 6 by execution of start-up instructions stored in the mask
12 ROM 22a (Figure 4).

13 In step 601, the validity of the stored program instructions (e.g., an Internet browser)
14 is checked to ascertain whether a corrupted state exists in the computer instructions stored
15 on the client 1 using a step of automatically checking. These computer instructions may be
16 stored, for example, on the flash memory 22b of the client system 1. This step of
17 automatically checking may be performed by the client system 1 in response to software,
18 hardware, or a combination of both. Any of a variety of routines may be used to check the
19 validity of the stored instructions such as, for example, a conventional checksum technique.

20 If the contents are valid (yes in decision block 602), then the normal start routine is
21 performed in step 603. If the contents of flash memory 22b are corrupted (no in decision
22 block 602), then an error download routine is performed (step 604). In the error download
23 routine, some or all of the corrupt information in the flash memory 22b is replaced by
24 correct information downloaded from the server 5. The error download routine is described

1 further below with respect to Figure 9 and Figure 10. Thus, the WebTV® system allows
2 errors in the programming or data to be detected and automatically corrected by performing
3 the error download routine, without intervention by the user of the WebTV® client system 1.

4 Figures 7 and 8 illustrate an optional extension of method 600 of Figure 6, whereby a
5 new version or upgrade of software is automatically made available to the client system.
6 Figure 7 depicts a routine 700 by which an upgrade of the Internet browser is initiated
7 during a normal start-up (e.g., step 603 of Figure 6). Initially, the client 1 automatically
8 connects to the WebTV® server 5 (step 701). Generally, this connection is made via the
9 modem pool 2 by executing a connection script. If an upgrade is determined to be available
10 (yes in decision block 702), and the upgrade is designated as mandatory (yes in decision
11 block 703), then the server 5 sends a command to the client 1 to cause a download request to
12 be written into the flash memory 22b of the client 1 (step 704). The client 1 is then
13 commanded by the WebTV® server 5 to reset (step 705). On the other hand, if an upgrade is
14 determined to be available (yes in decision block 702), but the upgrade is not designated as
15 mandatory (no in decision block 703), then the client 1 prompts the user to either accept or
16 decline the upgrade (step 706). If the upgrade is accepted (yes in decision block 707), then
17 the client 1 resets (step 705). If not (no in decision block 707), the routine ends.

18 Figure 8 illustrates portions 800 of the normal start-up routine for initiating an
19 upgrade. Initially, the client 1 determines whether a download request is present in flash
20 memory 22b (decision block 801). If not (no in decision block 801), the browser program is
21 started normally (step 808). If a download request is found in flash memory 22b (yes in
22 decision block 801), then it is next determined whether a connection script is present in flash
23 memory 22b (decision block 802).

1 As mentioned above, the connection script is executed to establish communication
2 with the WebTV® server 5 or the modem pool 2. Generally, a local connection script is
3 stored in flash memory 22b to allow connection to the local modem pool 2. A default
4 connection script is stored in mask ROM 22a, which allows direct connection to the
5 WebTV® server 5 using a toll-free (e.g., "1-800-") telephone number. The default
6 connection script is generally used only if a local connection script is not found in flash
7 memory 22b.

8 Referring still to Figure 8, if the local connection script is found in flash memory 22b
9 (Yes in decision block 802), then the client 1 uses that connection script to connect to the
10 WebTV® server 5 via the local modem pool 2 (step 803). Assuming such connection is
11 made, a determination is then made as to whether an IP (Internet Protocol) address, a port,
12 and path information for the upgrade is stored in the flash memory 22b (decision block 804).
13 The WebTV® server 5 may provide such information to the client 1 if the upgrade is to be
14 downloaded from a server other than a default server, which may be the WebTV® server 5.
15 If such information has been provided, then the client 1 connects to the specified server (step
16 805), requests the specified file (step 806), and initiates downloading of the file (step 807).
17 If an IP address, port, and path are not found in flash memory 22b (no in decision block
18 804), then the client 1 connects to the default server using a default IP address, port, and
19 path stored in mask ROM 22a (step 810). A default upgrade file is then requested using this
20 stored information (step 811). In step 811, the client 1 also indicates to the default server
21 which version of software it is currently running, so that the default server can determine the
22 proper default file for that client system. Downloading of the default file is initiated in step
23 807 after the client 1 has requested a file. As mentioned above, downloading to the client 1
24 occurs via the Internet 3 via the modem pool 2. Once downloaded, the information is

1 automatically decompressed (if compression was applied) and written into flash memory
2 22b by the client system 1.

3 Hence, an optional upgrade is performed automatically without any input from the
4 user of the client, other than the user's answering a prompt on whether to accept the upgrade.
5 A mandatory upgrade is performed without any input from the user and, in fact, can be
6 performed without informing the user, if desired.

7 While Figure 7 and 8 illustrate automatic upgrading of software as the client system
8 initiates a user session, the invention also extends to methods of upgrading software via a
9 satellite link. Such upgrading of software via a satellite link may be conducted
10 independently of any initiation of a user session and independently of any process for
11 checking for corrupted software. Using a satellite link, upgrades of software may be
12 initiated at any time that the server gains access to a new version of software. Figure 9
13 illustrates a method 900 whereby software is upgraded via the satellite link. Specifically,
14 whenever there is an upgrade of the browser software, the software is made available to
15 server 5 (step 901). If, according to decision block 902, client 1 is to receive the upgrade,
16 the server 5 downloads the upgrade replacement program instructions to the client 1 (step
17 903). This is possible since the server 5 establishes the satellite link 29' without the two-
18 way communication being established between server 5 and client 1. Thus, the client 1 need
19 not affirmatively do anything in order to receive the downloaded upgrade over the satellite
20 link 29' from the server 5.

21 The upgrade can be downloaded to client 1 at a time that client 1 is not engaging in
22 two-way communication with server 5. In other words, the upgrade can be received via the
23 satellite link at a time that client 1 has not logged on to server 5 and is otherwise not actively
24 directing communication to the server. Moreover, the upgrade can be received at a time that

1 client 1 is powered off. In this context, the term "powered off" refers to a state of the client
2 wherein features other than the ability to receive downloaded data from the satellite link are
3 not enabled. For instance, in a powered off state, client 1 can receive downloaded software
4 upgrades via that satellite, while being unable to browse the Internet or direct
5 communication to server 5. When client 1 received a software upgrade in a powered off
6 state, the upgrade is stored at the client so that it can potentially replace the software stored
7 at the client when the client is powered on.

8 An upgrade received according to the method of Figure 9 can optionally be
9 designated as mandatory or not mandatory by, for example, server 5. Designation of an
10 upgrade as mandatory or not mandatory has been previously described in reference to Figure
11 7. In the context of the method of Figure 9, which uses a satellite link to download
12 upgrades, the upgrades are transmitted to client 1 regardless of their mandatory or non-
13 mandatory status. Mandatory upgrades automatically replace the prior version of the
14 software at the client. Non-mandatory upgrades, however, only replace the prior version of
15 software after the user of client 1 has been prompted to accept the upgrade and after the user
16 has responded by providing user input indicating acceptance of the upgrade.

17 As stated above, the methods of Figure 7 and Figure 8 are optionally performed
18 when the validity of the stored program information is confirmed (yes in decision block 602
19 of Figure 6). However, if the step of automatically checking ascertains the existence of a
20 corrupted state in the stored program instructions (no in decision block 602 of Figure 6), the
21 client 1 automatically corrects the stored program instructions as is now described.

22 First, the client system 1 automatically connects to at least one of the remote servers
23 4 without the need for intervention from the user and, if desired, without informing the user
24 of the connection. Thus, embodiments within the scope of the present invention include a

1 means and step for automatically connecting the client system 1 to at least one of the server
2 systems 4 or 5 that contains a replacement for the stored program instructions of the client
3 system 1.

4 There are a variety of ways for connecting a client system to a server system. If the
5 location of the replacement program instructions is known, then a request can be
6 automatically generated and transmitted over the Internet 3 to the appropriate server 4 or 5.

7 Another example of a method of automatically connecting the client 1 to a server
8 that contains the replacement is represented by method 1000 of Figure 10. Initially, the
9 client 1 connects to the WebTV® server 5 directly using the default toll-free number stored
10 in mask ROM 22a (step 1001). Once connected, the client system 1 obtains a local
11 connection script from the server 5 (also step 1001). The client system 1 then disconnects
12 from the server 5 (step 1002) and then reconnects to the server 5 via the local modem pool 2
13 using the local connection script (step 1003). The client 1 then further establishes a
14 connection to the default server using the default IP address, port, and path stored in mask
15 ROM 22a (step 1004).

16 After the client system 1 has connected to the appropriate server 4 or 5 that contains
17 the replacement program instructions, the client 1 downloads the replacement program
18 instructions from the server that contains the replacement. Accordingly embodiments within
19 the scope of the present invention includes a means and step for automatically downloading
20 the replacement from the server that contains the replacement.

21 Figure 11 illustrates the downloading process 1100 in greater detail. To initiate
22 downloading, the client system 1 transmits a request to the default server (step 1101). For
23 example, the request may be for the replacement software from the default server using the
24 default IP address, port, and path. The client 1 also indicates to the server which version of

1 software it is currently running, so that the server can determine the proper default file for
2 that client 1.

3 In one embodiment, the client system 1 requests the replacement program
4 instructions by block number. In this case, the client system 1 receives certain block
5 information (step 1102) from the default server that will transmit the data, including the size
6 of the block to be downloaded, an identification of the next block to be downloaded, the
7 address in flash memory 22b to which the current block is to be written, and a description of
8 any data compression that is to be applied before transmission over the Internet 3. The data
9 is then transmitted over the Internet 3 to the client 1.

10 The client system 1 then receives the data (step 1103) via the network connection 29
11 and loads the data into RAM 23. The client system 1 decompresses the data (step 1104),
12 assuming data compression was used. The client 1 writes the decompressed data into flash
13 memory 22b (step 1105) using the address received in step 1102. If there is another block to
14 be downloaded according to the next block information (yes in decision block 1106), the
15 routine 1100 repeats from step 1101. Otherwise, the routine ends.

16 Thus, various methods according to the invention for upgrading or restoring software
17 stored at the client have been disclosed. Corrupted or inoperable software can be restored
18 automatically when a client is powered on by receiving replacement software from the
19 server. Outdated software can be replaced at any time by receiving at the client a download
20 of a software upgrade via a satellite link. The methods of restoring corrupted software and
21 upgrading outdated software can be practiced in combination or separately as desired.

22 The present invention may be embodied in other specific forms without departing
23 from its spirit or essential characteristics. The described embodiments are to be considered
24 in all respects only as illustrative and not restrictive. The scope of the invention is,

1 therefore, indicated by the appended claims rather than by the foregoing description. All
2 changes which come within the meaning and range of equivalency of the claims are to be
3 embraced within their scope.

4 What is claimed and desired to be secured by United States Letters Patent is:
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24